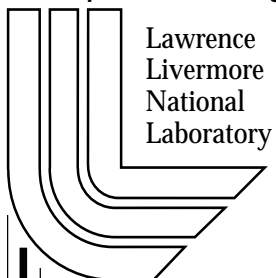


Some Aspects of Non-Split Vlasov Simulation Methods

A. Friedman, E. Sonnendrucker

August 7, 2002

U.S. Department of Energy



DISCLAIMER

This document was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor the University of California nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or the University of California. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or the University of California, and shall not be used for advertising or product endorsement purposes.

This is a preprint of a paper intended for publication in a journal or proceedings. Since changes may be made before publication, this preprint is made available with the understanding that it will not be cited or reproduced without the permission of the author.

Some aspects of non-split Vlasov simulation methods*

Alex Friedman¹ and Eric Sonnendrucker²

¹*LLNL and LBNL*

²*U. Strasbourg and LBNL*

(Dated: August 7, 2002)

This note summarizes some of the work done on Vlasov simulation methods during the two-week visit of E. Sonnendrucker to LBNL in July-August, 2002. It describes a variety of algorithms which might be employed in solving the coupled Vlasov-Poisson system on a grid in phase space. In particular, it emphasizes a class of methods which do not employ the sort of operator splitting commonly used in Vlasov methods, whereby the distribution function information is alternately updated along planes of constant x and constant v . Instead, the distribution function information on the phase space nodes is given by either (a) multi-dimensional interpolation of the gridded f at an earlier time onto a set of phase space locations which are obtained by tracking the nodes back to that earlier time (“backward semi-Lagrangian”), or (b) weighted deposition or averaging of f values from orbits onto nodes (“forward semi-Lagrangian” or “fully Lagrangian”).

I. INTRODUCTION

Vlasov methods and particle-in-cell (PIC) methods each have advantages and disadvantages. PIC methods have been the work-horses of both plasma and beam simulation. However, they can suffer from statistical “noise” because they, in essence, compute $\rho(x)$ by counting particles in cells (“nearest grid point”) or by means of a smoother but related procedure (“cloud in cell,” etc.). For some applications this noise is a significant consideration, mandating the use of large numbers of simulation particles. A Vlasov code tends to be less noisy but may suffer from inaccuracies and/or diffusion, especially when structures too fine to be well-resolved by the phase space grid arise in the course of the computation. Another aspect of the Vlasov method is its dynamic range: phase space regions of low density are tracked with the same fidelity as those of high density. This can be either desirable or undesirable, depending upon the application. It does imply a certain inefficiency in that effort is devoted to computing the charge density in regions which exert little influence on the self-field, in contrast with PIC where a type of “importance weighting” is automatically present.

The tradeoff is cost, especially in higher-dimensional systems. Indeed, for many problems PIC methods produce a sufficiently accurate answer at far lower cost than a Vlasov method. However, in beam physics problems where dynamic range is important (and especially in studies of beam “halo” formation), the process is sufficiently rapid that a simulation of a relatively short section of the beam line (for example, in the neighborhood of a transition in beam line parameters) is all that is needed. In such applications Vlasov methods may enable more efficient computations, though some further development is needed (*e.g.*, a time-varying phase space grid is needed for alternating gradient systems).

This note first discusses the various methods by which the charge density can be computed in a calculation based on semi-Lagrangian or Lagrangian methods. It

then describes a number of algorithms with comments on some aspects of the implementation.

II. METHODS OF CHARGE DEPOSITION

It is possible to compute the spatial charge distribution from a set of simulation particles or markers in several ways. The most familiar of these is the PIC deposition, whereby each simulation “superparticle” carries a preassigned total charge corresponding to that of a large number of physical particles (this number is often referred to as a “weight”), and a mass such that the charge-to-mass ratio, and thus yields the orbit, are correct. Most commonly, all superparticles are assigned the same weight, but this need not be the case. When variable weights are used (for example, to concentrate the particles in a region where improved statistics are desired), care must be employed; in some cases the “heavy” and “light” particles will mix, possibly leading to severe fluctuation problems. These arise if, *e.g.*, a heavy particle moves into a low-density region while light particles move into a high-density region and so fail to exert much influence on the local density.

In what we term a “Vlasov” prescription for the charge density $\rho(x)$, the local density at each spatial grid point is obtained by an integration of the multi-dimensional particle distribution function over velocities; with q the particle charge this is:

$$\rho(x) = q \int f(x, v) dv \quad (1)$$

This terminology is our own. Indeed, some “Vlasov” codes (*e.g.*, Denavit’s [1]) have used PIC deposition. Furthermore, the “ δf ” methods [3–8] commonly use PIC deposition. We consider such methods to be a class of Lagrangian Vlasov methods because either (a) the value of f is carried along the orbits (characteristics) and δf is computed as the difference between the f which is advected unchanged (per Liouville) along the marker orbit

and the locally-evaluated value of f_0 (as advocated by Aydemir [5] and by Friedman *et.al.* [8]), or (b) the perturbation δf (or a quantity related to it) is advanced along the marker orbit via an ordinary differential equation.

Most Vlasov codes indeed use some form of “Vlasov” prescription. In such a formulation, computation of $\rho(x)$ requires that, for each marker, x and v be known at the same time level. This stands in contrast with PIC methods in which $\rho(x)$ can be computed from the marker locations alone, and is a significant complication in algorithm design (for example, it precludes use of the simple leapfrog method commonly employed in PIC codes). The charge-deposition implementation differs significantly among the Vlasov methods:

(a) When a backward semi-Lagrangian method is used, all the markers are at the nodes when it is time to compute $\rho(x)$, and a suitably-normalized sum over nodal f values at the set of nodes associated with each spatial position yields ρ at that position.

(b) When a forward semi-Lagrangian method is used, the markers wind up at arbitrary locations within the phase space cells at the end of the time step. However, they have been “launched” from a uniform grid, and so each represents the same quantity of phase space volume. Thus, each may be thought of as containing an amount of charge proportional to its value of f . This charge can be weighted to the phase space grid using methods analogous to those employed in PIC for weighting to a position-space grid, such as an area-weighted scatter-add. Since the motion of the markers is a Liouvillean flow, over short times the phase space “cubes” may distort but the density of markers must remain constant (in the limit of perfectly-computed orbits). So, within an overall constant, a nodal f value obtained by a sum over marker contributions is approximately equivalent to an average over marker f values in the vicinity of that node. The expression used is:

$$f_k = \sum W_{jk} f_j \quad (2)$$

where W_{jk} is the particle “shape” function (for example, the area-weighting function) that defines the contribution of particle j onto node k based upon their phase space locations. This method can break down if the steps are too large, since the phase-space volumes may distort so severely (stretch along one axis and contract on others) that some nodes are left with no nearby markers, or with a smaller number than would be consistent with constant marker density.

(c) A “hybrid” method is based on the forward semi-Lagrangian technique, but with the marker identities preserved over multiple time steps. (A similar technique was used by Denavit, but he used a PIC-like deposition). Provided the markers are initially loaded onto a uniform grid in phase space, the situation is similar to that immediately above, and again the markers must be

re-launched before the phase space volumes associated with them distort too severely. Of course, a number of markers greater than the number of nodes might be employed, and with the correspondingly smaller phase space volumes associated with them it would take longer for non-linear distortions to invalidate the constant marker density assumption. Such a procedure would reduce the diffusion associated with re-launching markers, but might be an expensive remedy.

(d) A “Monte-Carlo” Vlasov method is also possible and has some attractive features. Here, the markers are placed in phase space where desired, not necessarily on a regular grid; they might be concentrated in regions where the initial f is varying rapidly, for example. Each carries its own value of f forever, but their associated quantities of phase space volume will, in general, differ. In such a system it is necessary to obtain the value of f at node k by *averaging over* the contributions of the markers in its vicinity, rather than by *summing* the contributions. The expression used is:

$$f_k = \frac{\sum W_{jk} f_j}{\sum W_{jk}} \quad (3)$$

An attractive aspect of this method is the prospect for introducing new markers or deleting old ones as the calculation evolves. This is relatively straightforward because the markers carry only information about the values of f , rather than a fixed amount of charge as in PIC. (It is challenging in PIC to add or delete particles because charge and momentum must be taken from old particles if they are to be given to new ones). However, it remains to specify a prescription for adding and deleting markers. Also, in regions where the spacing between markers is large, it would be desirable to employ a “wider” particle shape. Both of these concerns are automatically addressed by the wavelet-based Adaptive Mesh Refinement method, which also concentrates resolution where it is needed while naturally discarding fine-grained features which can’t be resolved even on the finest grid.

Computation of the quantity $\sum W_{jk}$ is a useful integrity check in all of the non-split methods discussed here; when it deviates significantly from its nominal value anywhere in the grid, the discrete analog of Liouville’s theorem is not being preserved, and so smaller step sizes and/or smaller grid cells are warranted. Alternatively, renormalization of the charge densities by this denominator might be employed in methods (b) and (c), just as in (d), to possible advantage. Split methods, which are based on separate advection steps along x and v lines, automatically conserve the volumes of the phase space “cubes” and so there is no need to compute such a check.

III. BACKWARD SEMI-LAGRANGIAN WITH SPLIT-LEAPFROG ADVANCE

In this method all markers start on the nodes and are tracked backward in time to their own particular phase

space locations at the earlier time level. The distribution function f_{grid} is obtained by interpolating the nodal f values onto the marker locations. This interpolation may be via splines, Hermite polynomials, or a particle shape function as in PIC, *e.g.* area weighting. The algorithm is shown in Fig. 1. In Steps 1-3 a first estimate of E at the

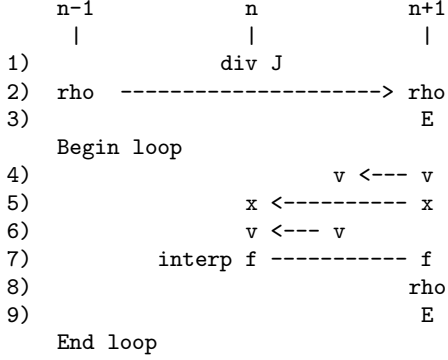


FIG. 1: Backward semi-Lagrangian with split-leapfrog advance.

advanced time level $n + 1$ is obtained; some options for this are discussed below. In Step 4 the marker velocities are pushed backward from the nodes to the (temporary) half time level $n + 1/2$ using the best-available estimate for E at time level $n + 1$, evaluated at the nodes. In Step 5 the marker positions are pushed backward using the newly computed temporary velocities. In Step 6 the velocities are pushed back to the earlier time level n using the (saved) nodal E at that time level, interpolated onto the newly computed positions. In Step 7 the distribution function values at the marker locations in phase space at time n are computed by interpolation; by Liouville's theorem these are also the new nodal f values. That is,

$$f^{n+1}(\text{nodes}) = f^n(x^n, v^n).$$

On Step 8 the nodal f values are summed over the velocity coordinates to obtain the charge density $\rho(x)$, and in Step 9 Poisson's equation is solved to obtain an improved nodal E field.

The algorithm was derived by thinking in terms of an “isochronous” variant of the leapfrog advance where x and v each begin and end every step on integer time levels. Nonetheless it is useful (and algebraically equivalent) to write this marker advance with right-hand sides as averages of old and new time levels, as follows:

$$\frac{x_i - x^{n-1}}{\Delta t} = \frac{v_j + v^{n-1}}{2}; \quad (4)$$

$$\frac{v_j - v^{n-1}}{\Delta t} = \frac{E^n(x_i) + E^{n-1}(x^{n-1})}{2}. \quad (5)$$

The predictor-corrector algorithm described above should work well. It is possible to perform further corrector iterations in a loop, as indicated in the figure. Convergence of such an iteration may be expected to be rapid.

The first estimate of E at the advanced time level $n + 1$ may be obtained in a number of ways:

- time-advancing $\rho(x)$ on the grid, using the relationship $\frac{\partial \rho}{\partial t} = -\nabla \cdot J$ obtained by taking the divergence of Ampere's law (this is just the continuity equation) and then solving the Poisson equation; this option is the one shown in Fig. 1.
- using a forward advance of the markers from the last step (a half-step in v followed by a full step in x) to yield a PIC-like $\rho(x)$ and then solving the Poisson equation.
- using a backward semi-Lagrangian step based on a leapfrog advance of double the usual step size (so that E^n , which is known, can be used for a time-centered backward velocity push), as described in Sec. V.
- using E from the last time step, or extrapolated from multiple earlier time steps (a linear, parabolic, or cubic fit to past data).

IV. FORWARD SEMI-LAGRANGIAN WITH SPLIT-LEAPFROG ADVANCE; “F” METHOD

In this method all markers start on the nodes and are advanced to their own particular phase space locations. The distribution function f_{grid} is obtained by weighting the f values carried along particle orbits onto the nodes using a “scatter-add” process and a particle shape function W of size roughly that of one cell; see option (b) of Sec. II. Again the first estimate of E may be obtained by various means. The algorithm is shown in Fig. 2. It is

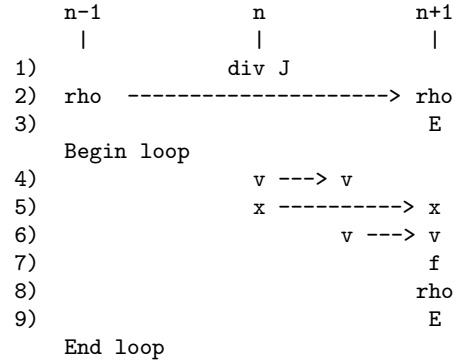


FIG. 2: Forward semi-Lagrangian with split-leapfrog advance.

possible to carry the markers forward over multiple steps, without re-launching them anew at each step. This might be done for some number of steps and then the markers re-launched. As mentioned in Sec. II, Denavit employed a similar technique but with a PIC-like charge deposition, so that his method reduces to PIC when the interval between re-launchings is made infinite. When the

markers are preserved indefinitely but a Vlasov prescription for ρ is used, we obtain a fully-Lagrangian method that we term the "F" method. It removes the diffusion of f associated with interpolations and re-launching markers. In principle it can be made both time-reversible and symplectic, though (in contrast with PIC) this seems to require iteration via a method such as the one shown above. However, ultimately as fine-grained structures in phase space (such as "spiral arms") develop, the charge density and field will become inaccurate, leading to inaccurate trajectories and perhaps to "collisional" effects similar to those in PIC methods (even if the method is made symplectic). So, to run an "F" method for a long time, one needs to use a large number of markers, or to introduce new ones, again as described in Sec. II. Thus, depending upon the problem, periodic re-launching (each step, or less frequently) may help control the formation of under-resolved structures.

V. BACKWARD SEMI-LAGRANGIAN BASED ON DOUBLE-STEP ADVANCE

It is possible to track x and v backward in a centered manner with a single field solution. At time step n , the algorithm advances f on the grid from time level $n-1$ to time level $n+1$; it is a backward semi-Lagrangian method because at time level $n+1$ the markers are located at the nodes. The basic equations to be solved are:

$$\frac{x_i - x^{n-1}}{2\Delta t} = v^n = \frac{v_j + v^{n-1}}{2}; \quad (6)$$

$$\frac{v_j - v^{n-1}}{2\Delta t} = E^n(x^n) = E^n\left(\frac{x_i + x^{n-1}}{2}\right). \quad (7)$$

Subtracting the latter from $1/\Delta t$ times the former yields

$$\frac{x_i - x^{n-1}}{2\Delta t^2} - \frac{v_j}{\Delta t} = -E^n\left(\frac{x_i + x^{n-1}}{2}\right). \quad (8)$$

An iterative procedure may be used to solve for x^{n-1} . We define:

$$x_m^{n-1} = x_{m-1}^{n-1} - \alpha \quad (9)$$

where m denotes the iteration level and we set $x_0^{n-1} = x_i$, that is, the grid coordinate. Expanding E^n about the "best-guess" location $(x_i + x_{m-1}^{n-1})/2$, we have:

$$\begin{aligned} \frac{x_i - x_{m-1}^{n-1} + \alpha}{2\Delta t^2} - \frac{v_j}{\Delta t} = \\ -E^n\left(\frac{x_i + x_{m-1}^{n-1}}{2}\right) + \frac{\alpha}{2}\nabla E^n\left(\frac{x_i + x_{m-1}^{n-1}}{2}\right). \end{aligned} \quad (10)$$

Solving for α and inserting the result into Eqn. 9, we obtain the iteration which is to be carried out until convergence is achieved:

$$x_m^{n-1} = x_{m-1}^{n-1} - \frac{2\left[\frac{v_j}{\Delta t} - E^n\left(\frac{x_i + x_{m-1}^{n-1}}{2}\right) - \frac{x_i - x_{m-1}^{n-1}}{2\Delta t^2}\right]}{\left[\frac{1}{\Delta t^2} - \nabla E^n\left(\frac{x_i + x_{m-1}^{n-1}}{2}\right)\right]}. \quad (11)$$

If linear interpolation is employed and ∇E computed using the corresponding expression, the iteration should converge almost immediately provided that the particle stays within the cell. The initial guess $x_0^{n-1} = x_i$ implies a gradient of E calculated at the node itself, so a centered two-cell difference of E may be needed to begin with.

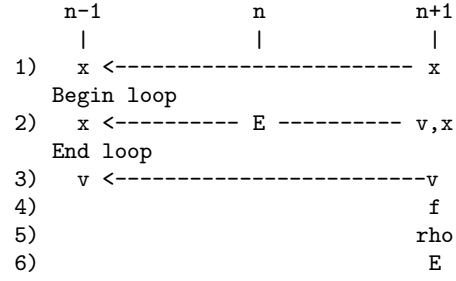


FIG. 3: Backward semi-Lagrangian based on double-step advance.

A graphical depiction of this algorithm is shown in Fig. 3. In Step 1 a first estimate is made of the "old" position x^{n-1} (which is needed so that the electric field E^n can be evaluated at the right position). Step 2 is the iteration loop described above, that is, the repeated application of Eqn. 11. Step 3 is Eqn. 7 rearranged to solve for v^{n-1} . Finally Steps 4-6 are the mapping of f , computation of $\rho(x)$, and Poisson solution.

Note that one does not actually "skip" any time steps in this algorithm; but the "latest" f , f^n , is not directly mapped onto the new f^{n+1} . This algorithm is appealing because it does not require repeating the Poisson solution during the step; and the iteration for x^{n-1} may be expected to converge rapidly. However, there is reason to be concerned about an undesirable "computational mode" arising because the distribution function at time level n is decoupled, as noted just above. An averaging procedure might be applied intermittently to mitigate such an occurrence.

Acknowledgments

*This work was performed under the auspices of the U.S. Department of Energy by the University of California, Lawrence Livermore and Lawrence Berkeley National Laboratories under Contract Nos. W-7405-Eng-48 and DE-AC03-76SF00098.

-
- [1] J. Denavit,
 - [2] E. Sonnendruker,
 - [3] W. M. Fawley, BAPS 27, 1034 (Oct. 1982).
 - [4] S. E. Parker and W. W. Lee, Phys. Fluids B 5(1), 77-86 (1993).
 - [5] A. Y. Aydemir, Phys. Plasmas 1(4), 822-31 (1994).
 - [6] G. Hu and J. A. Krommes, Phys. Plasmas 1(4), 863-74 (1994).
 - [7] P. H. Stoltz, R. C. Davidson, and W. W. Lee, Phys. Plasmas 5, (1998).
 - [8] A. Friedman, J. J. Barnard, and D. P. Grote, "Techniques for robust nonlinear delta-f simulations of beams," Proc. ICAP '98.